RENESAS Tool News

RENESAS TOOL NEWS on February 16, 2016: 160216/tn5

Note on Using the Following Tools: CS+ Code Generator for RL78 (CS+ for CC), CS+ Code Generator for RL78 (CS+ for CA and CX), e2 studio Code Generator Plug-in, Applilet3 Coding Assistance Tool for RL78

When using the CS+ Code Generator for RL78 (CS+ for CC), the CS+ Code Generator for RL78 (CS+ for CA and CX), the e2 studio (Code Generator Plug-in), or the Applilet3 coding assistance tool for RL78, take note of the problems on the following points that are described in this note.
1. Using the error interrupt of serial array unit 4 as UART4 or DALI4
Applicable products: RL78/I1A group
2. Using serial array unit 4 as DALI4
Applicable products: RL78/I1A group

---

1. Using the Error Interrupt of Serial Array Unit 4 as UART4 or DALI4

1.1 Applicable Products
- V2.03.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CC)
- V2.03.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CA,CX)
- V2.1.0.21 and later versions of the e2 studio (V1.0.0 and later versions of the Code Generator Plug-in)
- V1.00.00 and later versions of the Applilet3 for RL78

1.2 Applicable MCUs
RL78/I1A group

1.3 Description
Since the generated code has an error when error interrupts from serial array unit 4 as UART4 or DALI4 (digital addressable lighting interface) are selected, errors cannot be detected.

1.4 Workaround

Modify functions in r_cg_serial_user.c in the way shown below.
These modifications are required every time code is generated.

(1) In usage as a UART4

Modify the conditional expression shown below to judge from serial status register 41 (SSR41) if any among a framing error, parity error, or overrun error has occurred.

Before modification:

```
---------------------------------------------------------------------
__interrupt static void r_uart4_interrupt_receive(void)
{
    uint8_t rx_data;
    if (1U == SSR41)            /* Before modification */
    {
        r_uart4_interrupt_error();
    }
.............
}
---------------------------------------------------------------------
```

After modification:

```
---------------------------------------------------------------------
__interrupt static void r_uart4_interrupt_receive(void)
{
    uint8_t rx_data;
    if ((SSR41 & 0x0007UL) !=0)  /* Modify to judge raising of */
                                 /* the error-detected flag. */

    {
        r_uart4_interrupt_error();
    }
.............
}
---------------------------------------------------------------------
```

(2) In usage as a DALI4

Modify the conditional expression shown below to judge from serial status register 41 (SSR41) if any among a Manchester framing error, framing error, parity error, or overrun error has occurred.

Before modification:

```
---------------------------------------------------------------------
```

```
__interrupt static void r_dali4_interrupt_receive(void)
{
    uint8_t rx_data;
    if (1U == SSR41)            /* Before modification */
    {
        r_dali4_interrupt_error();
    }
.............
}
```

After modification:

```
__interrupt static void r_dali4_interrupt_receive(void)
{
    uint8_t rx_data;
    if ((SSR41 & 0x0087UL) !=0)  /* Modify to judge raising of */
                                 /* the error-detected flag.*/

    {
        r_dali4_interrupt_error();
    }
.............
}
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

2. Using Serial Array Unit 4 as DALI4

2.1 Applicable Products

- V2.03.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CC)
- V2.03.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CA,CX)
- V2.1.0.12 and later versions of the e2 studio (V1.0.0 and later versions of the Code Generator Plug-in)
- V1.00.00 and later versions of the Applilet3 for RL78

2.2 Applicable MCUs

- RL78/I1A group

2.3 Description

Since the generated code has an error when the length of units for

sending is set to 16 bits or the length for reception is set to 16, 17, or 24 bits, and serial array unit 4 is to be used as DALI4, transfer will not operate correctly.

2.4 Workaround

Modify functions in r_cg_serial.c in the way shown below.
These modifications are required every time code is generated.

(1) DALI sending with a bit length for transfer of 16 bits

Modify an operator from "&" to "%" as shown in the following example.

Before modification:

```
-----------------------------------------------------------------
MD_STATUS R_DALI4_Send(uint8_t * const tx_buf, uint16_t tx_num)
{
  MD_STATUS status = MD_OK;
  uint16_t tmp;
  if ((tx_num < 1U) || ((tx_num & 2U) != 0U))   /* Before */
                                        /* modification */
  {
    status = MD_ARGERROR;
  }
  else
  {
    ..............
  }
  ..............
}
-----------------------------------------------------------------
```

After modification:

```
-----------------------------------------------------------------
MD_STATUS R_DALI4_Send(uint8_t * const tx_buf, uint16_t tx_num)
{
  MD_STATUS status = MD_OK;
  uint16_t tmp;
  if ((tx_num < 1U) || ((tx_num % 2U) != 0U))   /* After */
                                        /* modification */
  {
    status = MD_ARGERROR;
  }
  else
  {
    ..............
```

```
  }
  ..............
}
-----------------------------------------------------------------
```

(2) DALI reception with a bit length for transfer of 16, 17, or 24 bits

Modify an operator from "&" to "%" as shown in the following example.

- Example of DALI reception with a bit length for transfer of 17 bits

Before modification:

```
-----------------------------------------------------------------
MD_STATUS R_DALI4_Receive(uint8_t * const rx_buf, uint16_t rx_num)
{
  MD_STATUS status = MD_OK;
  if ((rx_num < 1U) || ((rx_num & 3U) != 0U))   /* Before */
                                        /* modification */
  {
    status = MD_ARGERROR;
  }
  else
  {
    ..............
  }
..............
}
-----------------------------------------------------------------
```

After modification:

```
-----------------------------------------------------------------
MD_STATUS R_DALI4_Receive(uint8_t * const rx_buf, uint16_t rx_num)
{
  MD_STATUS status = MD_OK;
  if ((rx_num < 1U) || ((rx_num % 3U) != 0U))   /* After */
                                        /* modification */
  {
    status = MD_ARGERROR;
  }
  else
  {
    ..............
  }
..............
```

```
}
------------------------------------------------------------------
```

2.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.
